

UNITED STATES PATENT AND TRADEMARK OFFICE

VERIFICATION OF A TRANSLATION

I, Charles Edward SITCH BA,
Deputy Managing Director of RWS Group plc UK Translation Division, of Europa House,
Marsham Way, Gerrards Cross, Buckinghamshire, England hereby declare that:

My name and post office address are as stated below;

That the translator responsible for the attached translation is knowledgeable in the English language and in the Japanese language, and that, to the best of RWS Group plc knowledge and belief, the English translation of the marked portion of the attached Japanese document is true and complete.

I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true; and further that these statements were made with the knowledge that willful false statements and the like so made are punishable by fine or imprisonment, or both, under Section 1001 of Title 18 of the United States Code and that such willful false statements may jeopardize the validity of the application or any patent issued thereon.

Date: October 28, 2003

Signature : 

For and on behalf of RWS Group plc

Post Office Address :

Europa House, Marsham Way,
Gerrards Cross, Buckinghamshire,
England.

ITEM (For cited documents etc. refer to the List of Cited Documents etc.)

- Claim 1
- Cited Documents etc. 1
- Remarks

By virtue of the fact, using cited example 1, a 4-digit unit quotient is determined in a division operation, the divisor is scale-converted within a prescribed range, and a divisor that is multiplied by 3 is created, the invention pertaining to claim 1 can be easily devised from the invention described in cited example 1.

List of Cited Documents etc.

1. Japanese Unexamined Patent Application No. Heisei 07-248899

Record of Results of the Prior Art Documents Search

- Field searched IPC Edition 7 G06F7/52
- Prior Art Document
 Japanese Unexamined Patent Application No. 2000-10763

The record of the results of the prior art document search does not constitute a reason for rejection.

拒絶理由通知書

特許出願の番号 特願2000-029524
起案日 平成15年 9月26日
特許庁審査官 田中 友章 9376 5E00
特許出願人代理人 高橋 詔男(外3名) 様
適用条文 第29条第2項、第36条

この出願は、次の理由によって拒絶をすべきものである。これについて意見があれば、この通知書の発送の日から60日以内に意見書を提出して下さい。

理 由

[理由1]

この出願は、特許請求の範囲の記載が下記の点で、特許法第36条第6項第2号に規定する要件を満たしていない。

記

1. k を正の整数としたとき $1/2^k$ よりも $1/2^{k+1}$ のほうが小さいと認められるから、請求項1の「 $1/2^k$ 以上 $1/2^{k+1}$ 未満」という記載は意味が不明である。

よって、請求項1に係る発明は明確でない。

[理由2]

この出願の下記の請求項に係る発明は、その出願前日本国内又は外国において頒布された下記の刊行物に記載された発明に基いて、その出願前にその発明の属する技術の分野における通常の知識を有する者が容易に発明をすることができたものであるから、特許法第29条第2項の規定により特許を受けることができない。

記 (引用文献等については引用文献等一覧参照)

- ・ 請求項 1
- ・ 引用文献等 1
- ・ 備考

引用例1では除算において4桁単位に商を決定しており、除数を所定の範囲内にスケーリング変換しており、除数の3倍数を生成しているので、請求項1に係る発明は引用例1に記載された発明から当業者が容易に想到し得るものである。

この拒絶理由通知書中で指摘した請求項以外の請求項に係る発明については、現時点では、拒絶の理由を発見しない。拒絶の理由が新たに発見された場合には拒絶の理由が通知される。

引用文献等一覧

1. 特開平07-248899号公報

先行技術文献調査結果の記録

- ・調査した分野 I P C第7版 G 0 6 F 7 / 5 2
- ・先行技術文献 特開2000-10763号公報

この先行技術文献調査結果の記録は、拒絶理由を構成するものではない。

この拒絶理由通知の内容に関するお問い合わせ、または面接のご希望がございましたら下記までご連絡下さい。

特許審査第四部インターフェイス 田中友章

TEL. 03 (3581) 1101 内線3520

FAX. 03 (3580) 6907

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平7-248899

(43) 公開日 平成7年(1995)9月26日

(51) Int.Cl.⁶

G 0 6 F 7/49
7/52

識別記号

庁内整理番号

F I

技術表示箇所

C

3 2 0 N

審査請求 未請求 請求項の数 3 O L (全 7 頁)

(21) 出願番号 特願平6-42164

(22) 出願日 平成6年(1994)3月14日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目6番地

(72) 発明者 外村 元伸

東京都国分寺市東恋ヶ窪1丁目280番地

株式会社日立製作所中央研究所内

(74) 代理人 弁理士 小川 勝男

(54) 【発明の名称】 除算器

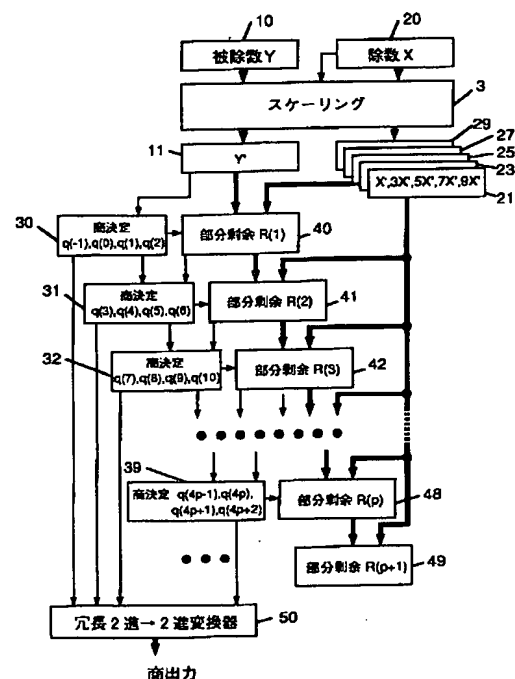
(57) 【要約】

【目的】 部分剰余計算に桁上げ伝播のない $\{-1, 0, +1\}$ の冗長2進加算器を利用する除算器において、部分剰余値の上位7桁で商を決定する効率的な除算器を提供する。

【構成】 除算器は、除数 X の範囲を $72/64 \leq X < 73/64$ にスケーリング変換する冗長2進加算器、桁借り先見付き減算器を設け、部分剰余計算に冗長2進加算器を設け、部分剰余値の上位7桁で商を決定することにより加算を制御する回路を設け、商を冗長2進→2進変換する回路を設けることにより達成される。

【効果】 従来、商1桁あたりの決定において、論理段数10段を必要としていたのに対して、4.75段で実現でき、約50パーセントの性能向上を提供できる。

図1



1

【特許請求の範囲】

【請求項1】 計算機システムの除算処理装置において、部分剰余計算に冗長2進数と呼ばれる $\{+1, 0, -1\}$ の表現を利用する冗長2進加算器を設け、除数 X を $72/64 \leq X < 73/64$ の範囲にスケーリング変換し、部分剰余値の上位7桁のみを参照することによって商決定を簡単にすることを特徴とする除算器。

【請求項2】 部分剰余計算において、次の結果の上位2桁分については、部分剰余値の上位7桁を参照して組み合せ回路によって直接求めることを特徴とする請求項1に記載した除算器。

【請求項3】 冗長2進加算回路の入力部に、 $+1$ と -1 の値が同時に入力されたときに、ゼロに相殺する機能を埋め込んだことを特徴とする冗長2進加算回路および該回路を採用した請求項1に記載した除算器。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、計算機システムの算術演算制御方式に係り、特に除算を高速に実現するのに好適な除算器に関する。

【0002】

【従来の技術】 一般に、除算は、1桁単位に商が決定され、その商決定にもとづいて部分剰余が計算され、次の演算ステップでは、この部分剰余値が1桁シフトされ、それにもとづいて次の桁の商が決定され、という具合に、繰り返し演算が実行される。この1桁単位に商を決定して行く方法は、基数2の除算法と呼ばれている。

【0003】 基数2の除算法では、 n 桁の商を求めるのに n 回の商決定と部分剰余計算の繰返し演算が必要である。従来、この繰返し演算数を $1/2$ に減らし、除算を高速化するために、2桁単位に商を決定していく基数4(=2の2乗)の除算法が知られている。例えば、電子情報通信学会英文論文誌A、1993年4月号の593~602ページのものがある(題名: シンプル クウォーシエント・ディジット・セレクション ラディックス4 ディバイダー ウィズ スケーリング オペレーション、著者: 外村元伸)。

【0004】 さらに除算を高速化するために、4桁単位に商を決定していく基数16(=2の4乗)の除算法が考えられるが、これに関しては、実際に効果のある効率的な方法はまだ考えられたことがない。しかも、部分剰余計算には、桁上げ伝播のない加算器、例えば、 $\{-1, 0, +1\}$ の冗長2進数表現を利用した冗長2進加算器を使用すると、総ゲート段数の少ない効率的な除算器が構成できるが、基数16で構成した例はまだ報告されていない。

【0005】

【発明が解決しようとする課題】 商の決定を冗長2進数表現の4桁単位に行い、4桁につき3桁分の部分剰余計算を削減し、1桁あたりの商の決定と部分剰余計算に必

2

要な総ゲート段数を基数2や4の場合よりも削減することにより高速な基数16の除算器を提供することにある。

【0006】

【課題を解決するための手段】 冗長2進数表現の4桁単位で商を決定することができるための条件を求めた結果、除数 X の範囲は $72/64 \leq X < 73/64$ (表記的には、 $[72/64, 73/64) = 1.001000 \dots$ と書く)で、7桁の参照で4桁単位に商を決定できることが解明された。しかも7桁解は最適かつ一意であり、この解以外は存在しない。従って、除数 X をこの範囲に収まるようにある数 M を掛けてスケーリング変換する。すなわち、正規化されている除数 $X = 0.1 \dots$ から、その小数点以下2桁目から8桁目の結果を参照し、冗長2進加算器と桁借り先見付き減算器を使用し、 $X' = MX = 1.001000 \dots$ のかたちに変換する。このとき、同時に被除数 Y に対しても同じ変換 M をかければ、商の値は変わらないので、正しく商が求まることになる。さらに、このとき、同時に、除数 X' の3倍、5倍、7倍、9倍値を求めておく。

【0007】

【作用】 本発明によれば、除数の範囲を $[72/64, 73/64)$ に収めるという余分なスケーリング変換が必要になるが、スケーリング変換のゲート段数が除算器全体に占める割合は1割程度なので、それよりも部分剰余計算の反復回数を削減する効果の方が大きいために高速除算器が実現できる。

【0008】

【実施例】 本発明の基数16の除算器の基本構成を図1に示す。被除数を Y 、除数を X とする。そして被除数 Y と除数 X の格納レジスタをそれぞれ10、20とする。これらは、スケーリング操作が回路3によって施されて、それぞれ $Y' 11$ 、 $X' 21$ に変換される。同時に、除数 X' の3倍(3 $X' 23$)、5倍(5 $X' 25$)、7倍(7 $X' 27$)、9倍(9 $X' 29$)もそれぞれ計算される。 Y' と X' の値にもとづいて、4桁単位に商が決定され(回路30)、冗長2進加算器を使って部分剰余計算が桁上げ伝播なしに実行される(回路40)。そして、部分剰余計算結果は4桁シフトされ、以下同様に、次々の桁の商が決定されていく。

【0009】 これらの演算回路30、31、 \dots ; 40、41、 \dots は、図1に示すように、配列状に配置される。最後に、決定された各桁の商 $q(-1)$ 、 $q(0)$ 、 $q(1)$ 、 $q(2)$ 、 \dots 、 $q(4p-3)$ 、 $q(4p-2)$ 、 $q(4p-1)$ 、 $q(4p)$ は冗長2進数で表現されているので、冗長2進から2進への変換器50によって通常の2進数に変換されて、最終的に商が求まる。

【0010】 以下、詳細に説明すると、まず、被除数 Y と除数 X はともに正規化されていて、 $0.1 \dots$ のかたちをしているものとする。これらは、図5に示すよう

3

に、除数Xの範囲を $[72/64, 73/64) = 1.001000 \dots$ に収めるというスケーリング変換Mによってそれぞれ Y'_{11} , X'_{21} に変換される。すなわち、図5に示すように、除数Xの小数点以下2桁目から8桁目の値を参照することによって変換し、 $X' = MX = 1.001000 \dots$ のかたちにする。図5で、1の上の傍線は-1であることを表す。以下でも同様の記述を使う。

【0011】図5に示したスケーリング変換Mを具体的
10 に行うために、図2に示す回路3が設けられる。回路3は、除数Xの小数点以下2桁目から8桁目の値にしたがってオペランド10, 20の値(図2では、j桁目の値を基準にj+1~j-6桁目)をシフト・セレクトする組み合わせ回路61, 62、冗長2進加算器(RB A. B)63、除数 X' の3倍(73), 5倍(75),

$$Q = Y/X = (MY)/(MX) = Y'/X'$$

基数16の除算は、数2の漸化式によって繰り返し演算実行される。

$$R(p+1) = 16 \cdot (R(p) - q(p) \cdot X')$$

ここで、pは演算の繰り返しステップ数を表し、小数点以下4p-1, 4p, 4p+1, 4p+2桁目の商を決定する演算に係わるものであることを示す。

【0016】R(p)はpステップ目の部分剰余計算を行う前の部分剰余値であり、この値にもとづいて小数点以下4p-1, 4p, 4p+1, 4p+2桁目の商が決定される。特に、 $R(0) = Y'$ である。そして、部分剰余が冗長2進加算器を使って桁上げ伝播なしで求められる。その部分剰余結果が16倍(4桁シフト)されて、次の演算ステップp+1で使われる部分剰余値 $R(p+1)$ になる。

【0017】図4は商数字選択回路を示す。部分剰余値 $R(p)$ の上位7桁($r(-1)$, $r(0)$, $r(1)$, $r(2)$, $r(3)$, $r(4)$, $r(5)$, : $r(j)$ は小数点以下j桁目の部分剰余値を表す)から商、 $q(4p-1)$, $q(4p)$, $q(4p+1)$, $q(4p+2)$ を決定する(図4の回路60)。

【0018】図6は図4に示す冗長2進加算器(部分剰余計算回路80の組み合わせで形成される)の加算を制御する信号出力の割当てを示す。同図の関数値mは X' のm倍数を選択することを意味する。図6は除数Xをスケーリング変換して X' を $[72/64, 73/64)$ の範囲に収めることによって求められたものである。符号信号sは、部分剰余値が非負のとき1で、負のとき0であり、加算制御信号の意味も兼ねている。

【0019】図7には、次の部分剰余結果の値の上位2桁を、冗長2進加算器を使わずに、直接組み合わせ回路で求めるための関数値を示してある。符号信号sは、やはり現在の部分剰余値が非負のとき1で、負のとき0である。sが負のときは、関数値の正負の符号が互いに入れ替わる。この組み合わせ回路によって、図6に示す引数値

4

7倍(77), 9倍(79)回路および桁借り先見付き減算器(BLB)81, 83, 85, 87, 89から構成されている。

【0012】冗長2進加算器(RB A. B)63は図3に示す1桁分の回路を必要桁分並べて実現される。入力側には、+1と-1表現部の信号がともに1のときには、+1-1=0だから、相殺してゼロにする回路が付加されている。この機能によって、初段の冗長2進加算器が不要になり、ゲート段数の節約になる。

【0013】 $X' = MX$, $Y' = MY$, 商をQとすると、次の数1なる関係にあるので、スケーリング操作Mによっても正しく商が求まることがわかる。

【0014】

【数1】

…(数1)

【0015】

【数2】

…(数2)

20 の上位2桁が先に確定され、商決定のための組み合わせ回路のゲート段数が節約される。なぜならば、もし、組み合わせ回路のかわりに冗長2進加算器で計算すると、引数値の上位2桁よりさらに上位桁に冗長2進数の非ゼロ値が発生し、これをゼロに還元するためにゲート段数を余分に消費するからである。

【0020】また、図7の関数値の組み合わせ回路は、図4に示した商決定と部分剰余計算回路のゲート段数以内に収まるだけの余裕をもっている。したがって、この組み合わせ回路はゲート段数を数える意味がないことから、
30 ここではその回路例を示すことを省略する。

【0021】

【発明の効果】本発明によれば、従来、商1桁あたりの決定において、基数2の除算器は論理段数10段を必要としていたのに対して、基数16の除算器は4.75段で実現できるため、約50パーセント以上の性能向上を提供できるという効果がある。ただし、基数16の除算器は、基数2のものに比べて2~3倍のゲート数を必要とし、しかも設計はかなり複雑になる。

【図面の簡単な説明】

【図1】本発明の一実施例の基数16の除算器の構成図。

【図2】本発明の一実施例のスケーリング変換器のブロック図。

【図3】本発明の一実施例のスケーリング変換で使用する相殺機能付き冗長2進加算回路(1桁分)。

【図4】本発明の一実施例の任意ステップの商決定とある桁の部分剰余を計算する回路。

【図5】本発明の一実施例のスケーリング変換を示す図。

【図6】冗長2進加算器の加算を制御する信号割当てを示

す図。

【図 7】部分剰余結果の値の上位 2 桁を組み合わせ回路で求めるための関数値を示す図。

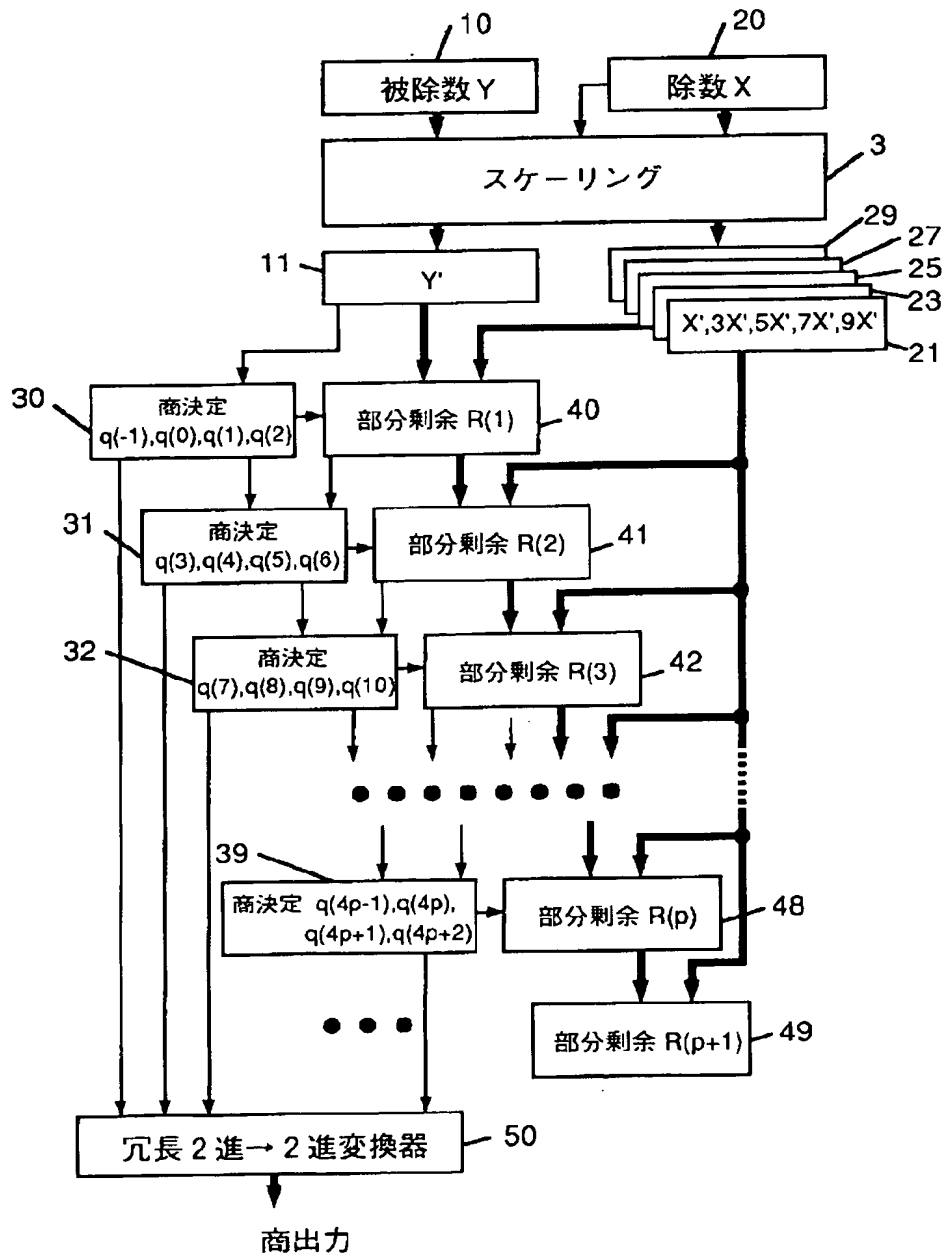
【符号の説明】

3…スケール変換回路、10…被除数 Y 格納レジスタ、20…除数 X 格納レジスタ、11…スケール変換された被除数 Y' 格納レジスタ、21, 23, 25, 27, 29…スケール変換された除数 X' の n 倍値

格納レジスタ、30, 31, 32, 39…商決定回路、40, 41, 42, 49…部分剰余計算回路、50…冗長 2 進から 2 進への変換回路、60…商決定回路、63…スケール変換ブロックで使われる相殺機能付き冗長 2 進加算回路、73, 75, 77, 79…n 倍加算を実行する冗長 2 進加算器、80…任意桁の部分剰余計算回路、81, 83, 85, 87, 89…桁借り先見付き減算器 (BLB)。

【図 1】

図 1



【図 2】

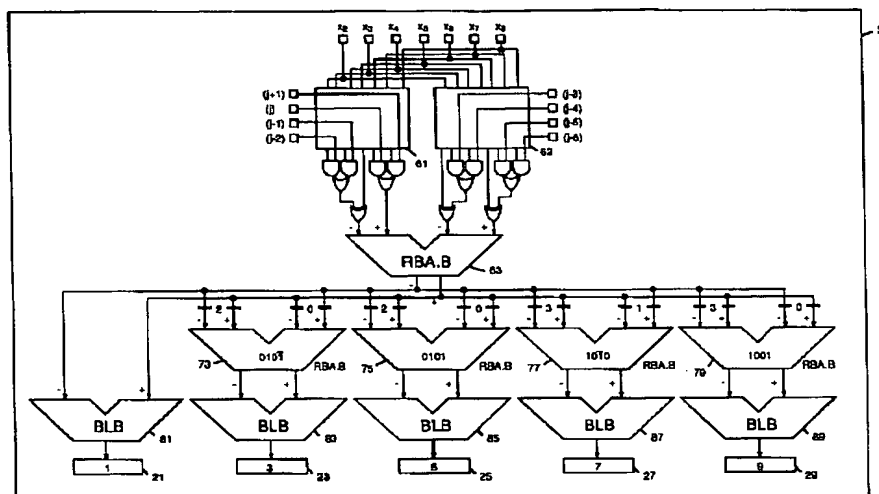
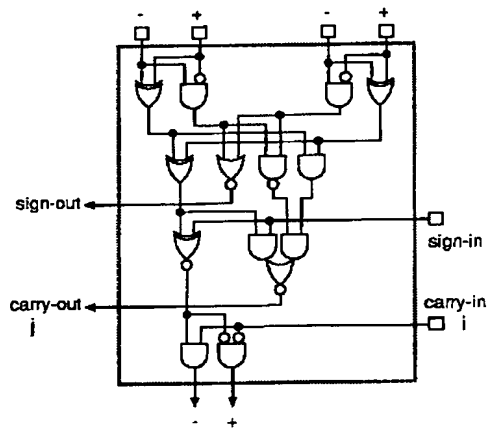


図 2

【図 3】

図 3



【図 6】

図 6

		$f_3 f_4 f_5$							
		000	001	010	011	100	101	110	111
$f_1 f_2 f_3$	000	0	0	0	0	0	0	1	1
	001	1	1	1	1	1	1	1	2
	010	2	2	2	2	2	2	2	2
	011	3	3	3	3	3	3	3	3
	100	3	4	4	4	4	4	4	4
	101	4	4	5	5	5	5	5	5
	110	5	5	5	6	6	6	6	6
	111	6	6	6	6	7	7	7	7
	1000	7	7	7	7	7	8	8	8
	1001	8	8	8	8	8	8	9	9
	1010	9	9	9	9	9	9	9	10
	1011	10	10	10	10	10	10	10	10

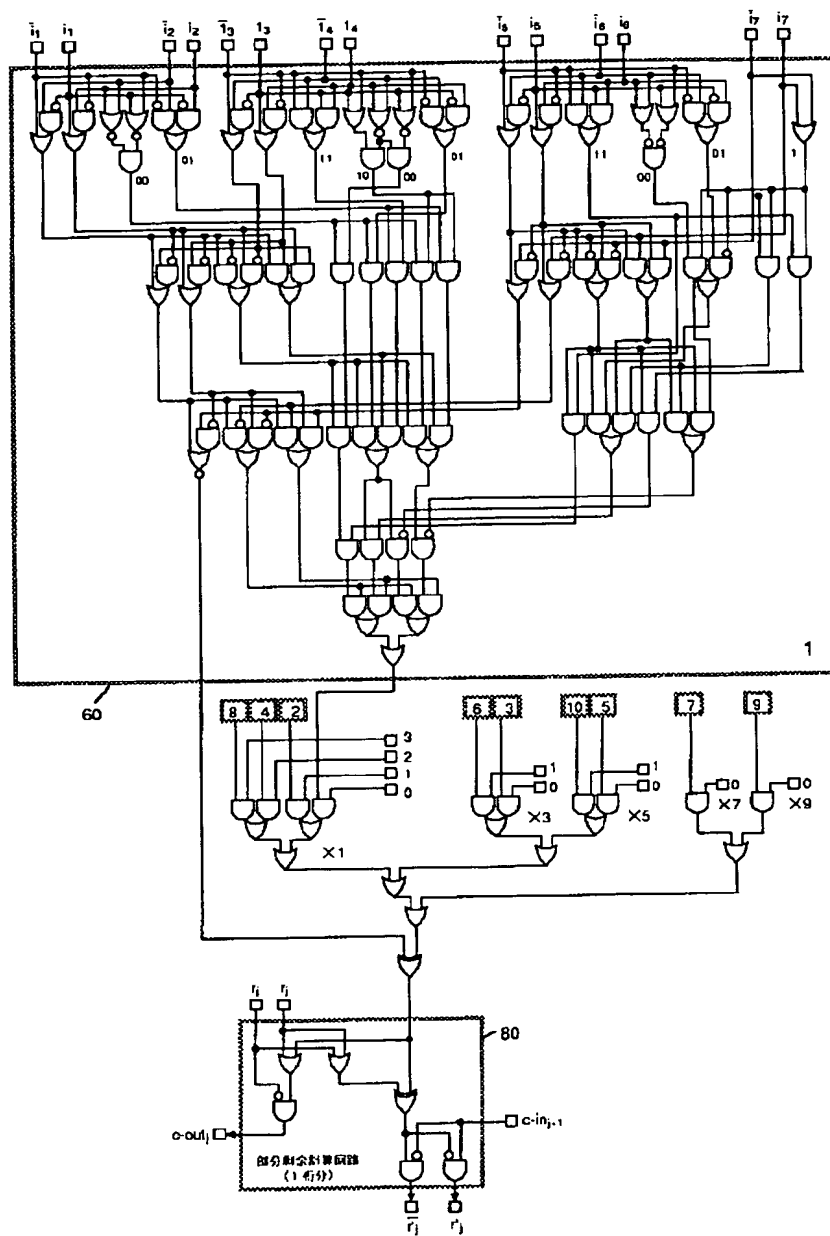
【図 7】

図 7

		$f_3 f_4 f_5$							
		000	001	010	011	100	101	110	111
$f_1 f_2 f_3$	000	00	00	01	01	10	10	10	01
	001	01	00	00	01	01	10	10	10
	010	01	01	00	00	01	01	10	10
	011	10	01	01	00	00	01	01	10
	100	10	10	01	01	00	00	01	01
	101	10	10	10	01	01	00	00	01
	110	01	10	10	10	01	01	00	00
	111	01	01	10	10	10	01	01	00
	1000	00	01	01	10	10	10	01	01
	1001	00	00	01	01	10	10	10	01/10
	1010	01	00/01	00	01/00	01	10/01	10	10
	1011	01/10	01	00/01	00	01/00	01	10/01	10

【図 4】

図 4



【図5】

図5

$$MX = [1.001000, 1.001001)$$

$x_1 \cdots x_8$	$m_1 m_0 \cdots m_6$	$x_1 \cdots x_8$	$m_1 m_0 \cdots m_6$	$x_1 \cdots x_8$	$m_1 m_0 \cdots m_6$	$x_1 \cdots x_8$	$m_1 m_0 \cdots m_6$
.10000000	10010000	.10100000	10010100	.11000000	10100000	.11100000	01010101
.10000001	10010001	.10100001	10010101	.11000001	10100001	.11100001	01010010
.10000010	10010010	.10100010	10010010	.11000010	10100001	.11100010	01010010
.10000011	10010101	.10100011	10010001	.11000011	10100001	.11100011	01010001
.10000100	10010100	.10100100	10010001	.11000100	10100010	.11100100	01010001
.10000101	10010101	.10100101	10010000	.11000101	10100010	.11100101	01010001
.10000110	10001010	.10100110	10010001	.11000110	10100101	.11100110	01010000
.10000111	10001001	.10100111	10010001	.11000111	10100101	.11100111	01010000
.10001000	10001000	.10101000	10010010	.11001000	10100100	.11101000	01010000
.10001001	10001001	.10101001	10010101	.11001001	10100100	.11101001	01010001
.10001010	10001010	.10101010	10010101	.11001010	10100100	.11101010	01010001
.10001011	10000101	.10101011	10010100	.11001011	10100101	.11101011	01010001
.10001100	10000100	.10101100	10010100	.11001100	10100101	.11101100	01010010
.10001101	10000101	.10101101	10010101	.11001101	10101010	.11101101	01010010
.10001110	10000010	.10101110	10101010	.11001110	10101010	.11101110	01010010
.10001111	10000001	.10101111	10101010	.11001111	10101001	.11101111	01010101
.10010000	10000000	.10110000	10101001	.11010000	10101001	.11110000	01010101
.10010001	10000000	.10110001	10101000	.11010001	10101000	.11110001	01010101
.10010010	10000001	.10110010	10101000	.11010010	10101000	.11110010	01010100
.10010011	10000010	.10110011	10101001	.11010011	10101000	.11110011	01010100
.10010100	10000101	.10110100	10101001	.11010100	10101001	.11110100	01010100
.10010101	10000100	.10110101	10101010	.11010101	10101001	.11110101	01010101
.10010110	10000101	.10110110	10101010	.11010110	10101010	.11110110	01010101
.10010111	10001010	.10110111	10100101	.11010111	10101010	.11110111	01010101
.10011000	10001010	.10111000	10100101	.11011000	10101010	.11111000	01010101
.10011001	10001001	.10111001	10100100	.11011001	01010101	.11111001	01001010
.10011010	10001000	.10111010	10100101	.11011010	01010101	.11111010	01001010
.10011011	10001001	.10111011	10100101	.11011011	01010100	.11111011	01001010
.10011100	10001001	.10111100	10100010	.11011100	01010100	.11111100	01001001
.10011101	10001010	.10111101	10100010	.11011101	01010100	.11111101	01001001
.10011110	10010101	.10111110	10100001	.11011110	01010101	.11111110	01001001
.10011111	10010100	.10111111	10100001	.11011111	01010101	.11111111	01001001